Goddard

LMSC Document No. F47064

278838

p. 48

FINAL REPORT

to
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD

NASA Contract Number NAS5-30514

EARTH'S EXTERNAL MAGNETIC FIELDS AT LOW ORBITAL ALTITUDES

Period: May 18, 1988 - May 18, 1990

Principal Investigator:

Dr. D. M. Klumpar

Lockheed Palo Alto Research Laboratory
Space Sciences Laboratory
Department 91-20, Building 255
3251 Hanover Street
Palo Alto, CA 94304

May, 1990

(NASA-CR-189255) EARTH'S EXTERNAL MAGNETIC FIELDS AT LOW ORBITAL ALTITUDES Final Report, 18 May 1988 - 18 May 1990 (Lockheed Missiles and Space Co.) 48 p CSCL 04A

N92-18118

Unclas

G3/46 0278838

EARTH'S EXTERNAL MAGNETIC FIELDS AT
LOW ORBITAL ALTITUDES

Final Report
Period: 18 May, 1988 to 18, May 1990

Contract NAS5-30514

Submitted by D. M. Klumpar
Lockheed Palo Alto Research Laboratories

INTRODUCTION AND BACKGROUND

Under our June, 1987 proposal entitled "Magnetic Signatures of Near-Earth Distributed Currents" we proposed to render operational a modeling procedure that had been previously developed to compute the magnetic effects of distributed currents flowing in the magnetosphere-ionosphere system. After adaptation of the software to our computing environment we would apply the model to low altitude satellite orbits and would utilize the MAGSAT data suite to guide the analysis. A contract to conduct this effort was awarded to LMSC in May, 1988.

During the first year basic computer codes to run model systems of Birkeland and ionospheric currents and several graphical output routines were made operational on a VAX 780 in our research facility. Software performance was evaluated using an input matchstick ionospheric current array, field-aligned currents were calculated and magnetic perturbations along hypothetical satellite orbits were calculated. The basic operation of the model was verified. Software routines to analyze and display MAGSAT satellite data in terms of deviations with respect to the earth's internal field were also made operational during the first year of effort. The complete set of MAGSAT data to be used for evaluation of the models was received at the end of the first year. A detailed annual report in May 1989 described these first year activities completely. That first annual report is included by reference in this final report.

This document summarizes our additional activities during the second year of effort under the contract, describes the modeling software and its operation, and includes as attachment the deliverable computer software specified under the contract.

MODEL DESCRIPTION

Description of Modeling Procedure

The modeling software described below is designed to facilitate studies of the contributions that ionospheric and magnetospheric currents make to the magnetic fields measured on low altitude polar orbiting satellites. Emphasis is placed on high latitude current systems because, as previously has been shown, large and highly variable perturbations of the geomagnetic field are associated with high latitude auroral phenomena. The routines compute the vector magnetic contributions at any point that arise from currents flowing in the ionosphere and along the magnetic field into the magnetosphere. The contribution due to a distant equatorial Ring Current is also included.

The basic modeling codes take, as input, an array of ionospheric currents distributed over a specified region of the polar ionosphere. Figure 1 illustrates the basic grid cell concept and shows the distribution of current locations for a coarse setup of 4 latitude cells and 12 longitude cells. This 4x12 array is illustrative for the purpose of showing how the grid cells, the ionospheric currents, and the Birkeland currents relate to one another. In actual use of the model a much finer grid consisting of many more grid cells would be employed. The view is looking down upon the polar regions with the earth's diple axis protruding from the center of the diagram. Magnitude and direction of the horizontal currents are specified for each point on a spherical surface at some altitude above the earth's surface. This altitude constitutes the base of the cell. The horizontal currents are shown in Figure 1 by arrows at the center of each grid cell. Each such horizontal current specification thereby locates a current cell, the bounds of which are determined by the density of current specification points. The latitude extent of the entire current system is bounded at the northern and southern boundaries by the first and last cell boundaries. In this illustration the latutude range is from 70° to 50°.

Currents flowing outward and inward along magnetic field lines provide the sources and sinks for the horizontal ionospheric currents. These field-aligned currents constitute the Birkeland current system. In practice these currents are represented by straight filaments which are tangent to the magnetic field lines at the ionosphere (the base of the cell) and are three earth radii in length. A more complex model, in which the field-aligned filaments curved with the magnetic field lines all the way to the equatorial plane was tested. The increased complexity made a barely discernable difference in the final result and because of this reason and in view of the increased computation time the simpler concept was adopted. Once the ionospheric currents have been specified and the region boundaries have been determined the modeling codes compute the Birkeland currents at each cell boundary as required for the preservation of current continuity. In Figure 1 these currents are shown as circles located at each cell boundary. Each circle encloses about 90% of the kurtically distributed current.

The magnetic field at any point is the vector sum of the contributions from all current elements in the system. Once the currents have been defined as described above the software can compute the magnetic field at any location. In practice, as will be described below, we compute the magnetic field along a satellite orbit through (and above) the current system so as to be able to relate the field signatures observed in real satellite data to the model fields.

A very important aspect of the model is the representation of the currents. The current elements are like current carrying wires, in that they have thickness. But unlike wires, they have smoothly varying cross-sectional current density. The current density in each element is platykurtically distributed over the cross-section of the current element. The kurtic representation is one which varies as the hyperbolic secant of the square of the distance from the center of the element. This gives a more realistic representation of the currents and prevents large and sudden transitions of the magnetic field.

Software Codes

The Fortran source codes revived under this contract and submitted as deliverables items are attached as Attachment B. The suite of Fortran source code to directly carry out the modeling and display consists of more than 1500 lines of code. The component codes are listed in the following table. Also shown in the table are the input and output data files.

Table 1: Magnetic Modeling Routines

INPUT	SOFTWARE CODE	OUTPUT
Run-time	SCURDIS	DIS.DAT
DIS.DAT + run-time	SAMPLT	Graphics
DIS.DAT + run-time	SBRKPLT	Graphics
DIS.DAT + run-time	SBRKALC	MAG.DAT
MAG.DAT + run-time	SBRKPLT	Graphics

A functional description of each code follows:

SCURDIS This Fortran routine sets up the current system with the input currents. The routine calculates all the necessary geometric parameters needed to specify the current elements and their location in space and writes these parameters along with the current magnitudes in the output file DIS.DAT. Runtime input information required for the code to run consists of specifying the number of cell rings, number of longitude sectors, inner and outer co-latitude boundaries of the current system, maximum current filament radius and latitude thickening exponent.

SAMPLT This is a graphical output routine that plots on a polar projection all Birkeland currents required to complete continuity with the specified ionospheric currents. The circlegram shows the magnitude and location of the Birkeland current filaments. Input data file is the DIS.DAT file produced by SCURDIS. Additional runtime input includes a divisor which controls the current represented by each line of the circlegram. Output is graphical and is delivered to one of nine possible output devices specified under runtime control. The format of this output will be dependent upon facilites at the operator site. Standard Tektronics 4010 and laser postscript output are among possible options.

SBRKPLT This is a graphical output routine that plots on a polar projection all horizontal ionospheric currents in the input system. The vectors show the location, magnitude and direction of each cell current filament. Input data file is the DIS.DAT file produced by SCURDIS. Additional runtime input includes a multiplier which controls the length of the vector in terms of the current magnitude. Output is graphical and is delivered to one of nine possible output devices specified under runtime control. The format of this output will be dependent upon facilites at the operator site. Standard Tektronics 4010 and laser postscript output are among possible options.

SBRKALC This routine calculates measurement positions for a specified satellite orbit and calls SMAGMOD. Input data file is the DIS.DAT file produced by SCURDIS. Additional runtime input includes the orbital altitude of the satellite, it's inclination and local time location of the orbit plane. The density of measurement points along the orbit is also controlled by runtime input. A switch is provided at input to allow the user to select only a part of the global currents to determine the individual contributions of various parts of the current system to the magnetic field. It is also possible to compute

low latitude magnetic perturbations. This routine calls SMAGMOD. Output is written to file MAG.DAT.

SMAGMOD This routine calculates, for each measurement location, the net vector magnetic field due to all of the distributed currents. It is called by SBRKALC. At each measurement point the vector sum of the individual contributions from each and every current element in the system is computed and returned to SBRKALC along with the magnitude of the Birkeland current at each measurement point.

SBRKPLT This is a graphical output routine that plots the three components of the magnetic perturbation at each measurement point along the satellite orbit specified in SBRKALC. Basic input array is provided by MAG.DAT. User runtime choices allow the displayed perturbations to be cast into one of four coordinate representations: XYZ, NEV, SDV, and ABZ. Output is graphical and is delivered to one of nine possible output devices specified under runtime control. The format of this output will be dependent upon facilites at the operator site. Standard Tektronics 4010 and laser postscript output are among possible options.

Running a Model

As an example of the model routines in actual use we illustrate in the following a complete run of the modeling system. The output is shown in Figures 2-4. A complete run through the system starts by running SCURDIS to set up the ionospheric input currents. In the version of SCURDIS contained in this report, up to 20 latitude cells and up to 24 longitude cells may be used. Each cell contains a specification for the vector ionospheric current at that location. Since Hall and Pederson conductivities are often used in physical descriptions of the auroral ionosphere and since ionospheric currents are often discussed in terms of the eastward and westward electrojects , we choose in practice to specify the ionospheric currents by their E-W and N-S components. The data file produced by SCURDIS contains information regarding the location, direction, and magnitude of the currents to be passed on to subsequent programs in the sequence.

Once the complete current specification has been set up by CURDIS, two plotting programs (SAMPLT and SCURPLT) are available to view the ionospheric and the resulting field-aligned currents. Figure 2 depicts the input ionospheric current system used for this example. Looking down upon the earth's polar region the figure shows, on a local time vs latitude polar projection, the ionospheric current vectors at each grid cell input point. This current system is the direct input for the model. In this model run a cell matrix of twenty latitude rings by 24 longitude sectors is used to represent an ionospheric current that has a strong eastward electrojet throughout the post noon sector and into the morning sector. The current is restricted to the latitude range from 57 to 85 degrees. This output is produced by the SCURPLT program.

The model calculates the field-aligned currents required to maintain current continuity. For this example the resulting field-aligned current distribution through an imaginary spherical shell above the ionospheric currents is shown in Figure 3. Field-aligned currents flow in and out at each grid point as required to maintain current continuity. Since there is a string divergence in the ionospheric currents along the noon meridian (see Figure 2) there is a strong downward field-aligned current at this location as seen in Figure 3. Current flows upward or downward along magnetic field lines as indicated by the horizontal or vertical hatching, respectively.

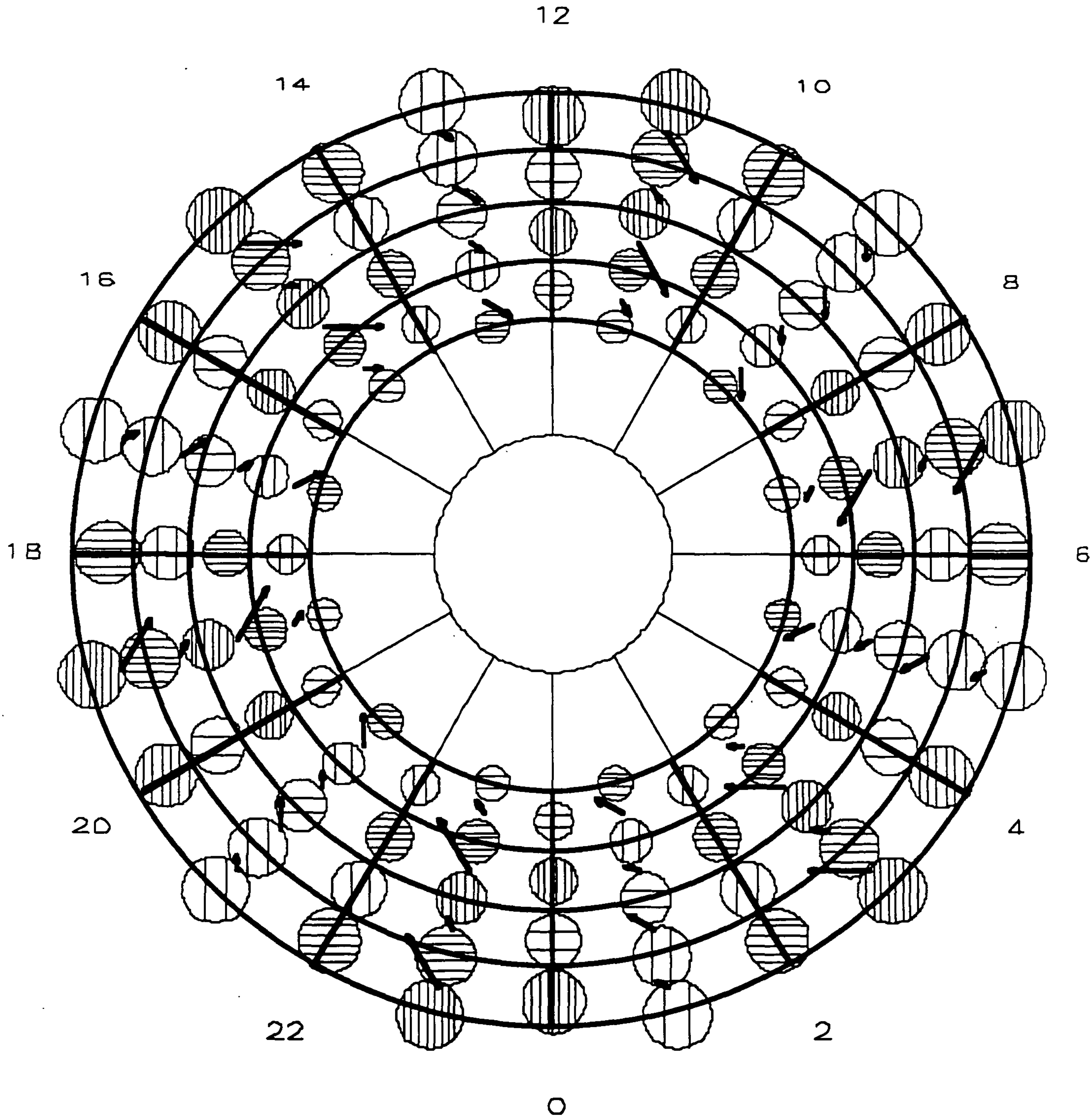
The product of ultimate interest is the magnetic field at any point. The magnetic perturbation code (SBRKALC) is set up to receive a specified satellite orbit and to

calculate the magnetic field perturbations at points along that orbit for one of several possible coordinate systems. SBRKALC is given the orbital parameters for the satellite and calculates the magnetic perturbations due to any one or all parts of the current system. The resulting information is written to the file MAG.DAT which in tern may be plotted using the program SBRKPLT. As an example, Figure 4 illustrates the magnetic field perturbations due to the model current system shown in Figures 2 and 3 that a satellite at 450 km altitude would see as it passes over the current system in an evening to morning orbit as shown in the upper right hand clock dial. Perturbations are shown for the three vector components northward (N), eastwardward (E) and vertical (V). The bottom panel shows the magnitude of the local field-aligned currents that the satellite passes through while the magnetic perturbations in the upper three panels are due to all of the currents in the entire system. The coordinate system for output of the perturbations is selected at runtime in BRKPLT.

Other activities during second year

An abstract was submitted, and accepted for presentation, at the spring 1990 American Geophysical Union meeting of results of the investigations carried out under this contract. The paper is entitled, "A Method for Computing Magnetic Perturbations at Satellite Altitude due to Distributed Currents in the Ionosphere and Magnetosphere". The paper emphasizes the capabilities of the model as a tool to examine the relative contributions of the magnetic perturbations which result from the various components of the current distribution. The capability to selectively "switch on" parts of the current system while evaluating the magnetic perturbations is a powerful tool for understanding the sources of the magnetic perturbations seen on earth-orbiting satellites. In the computer model there are four distinct current components: ionospheric N-S, ionospheric E-W, field-aligned, and ring current. Of course the currents themselves are intimately tied together by the requirement for current continuity and cannot be independently controlled without creating a physically impossible situation. The magnetic contribution, however, of each of the four can be selectively switched on to determine the effect of any one of them on the magnetic field at any measurement point. This provides a unique analysis tool that the natural environment cannot provide. It allows us to examine singularly the effect on the magnetic perturbations seen at the satellite due to each major component in the global current system. Such a diagnostic is an indispensable tool for understanding the behavior of the magnetic perturbations in orbit. A copy of the abstract is attached to this report as Attachment A.

Most of our progress with respect to the MAGSAT data was directed toward getting the data tapes and modifying our MAGSAT analysis routines to run on the VAX 780. The software has been transported to the VAX and is running successfully. Several figures in our first annual report showed sample plots from these analysis routines, and will not be repeated here.



DISTRIBUTION OF CURRENTS

FIGURE 1

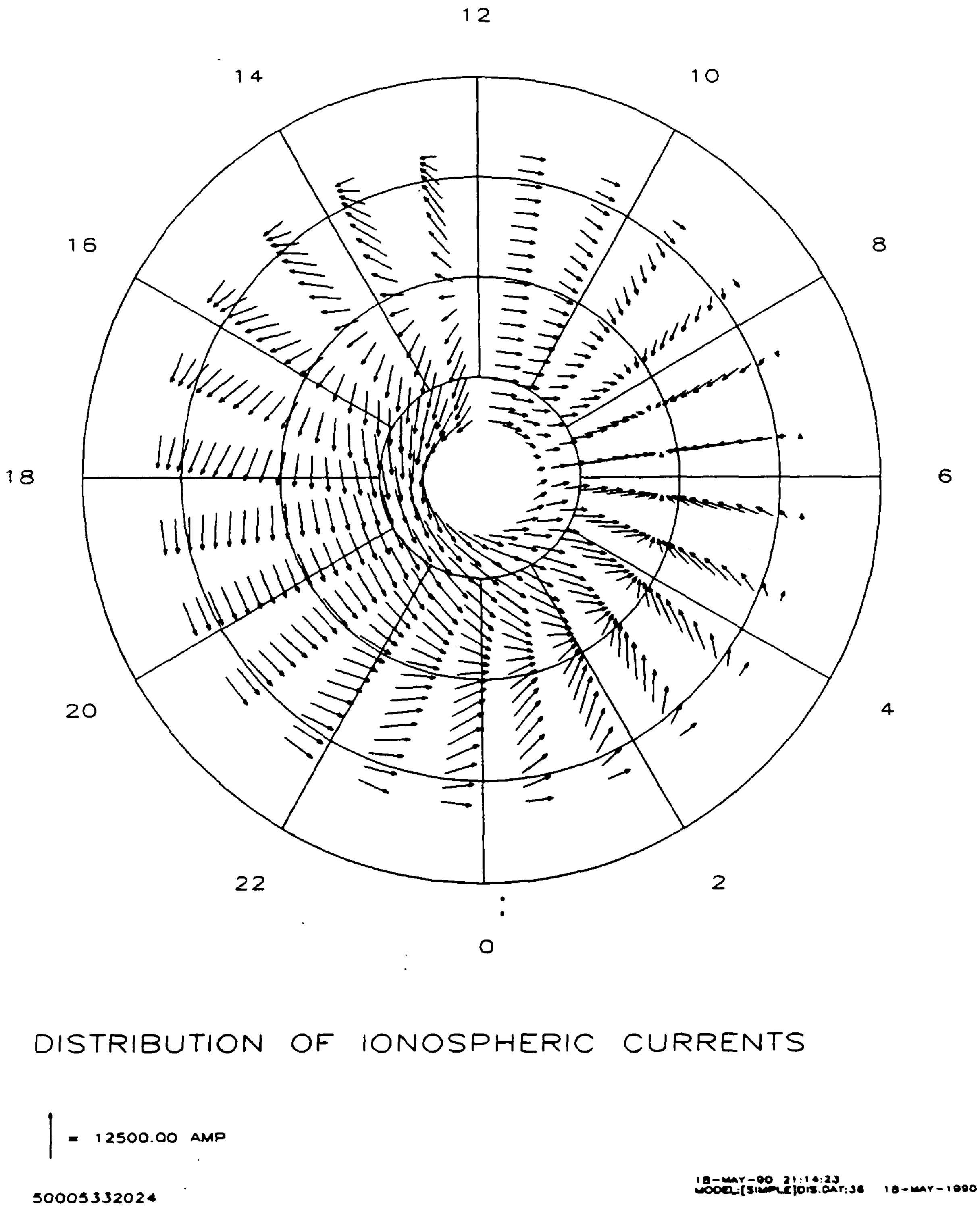

12
10
8
6
4
2
0
22
20
18
16
14
DISTRIBUTION OF IONOSPHERIC CURRENTS
= 12500.00 AMP
50005332024
18-MAY-90 21:14:23
MODEL:[SIMPLE]DIS.DAT:36 18-MAY-1990

FIGURE 2

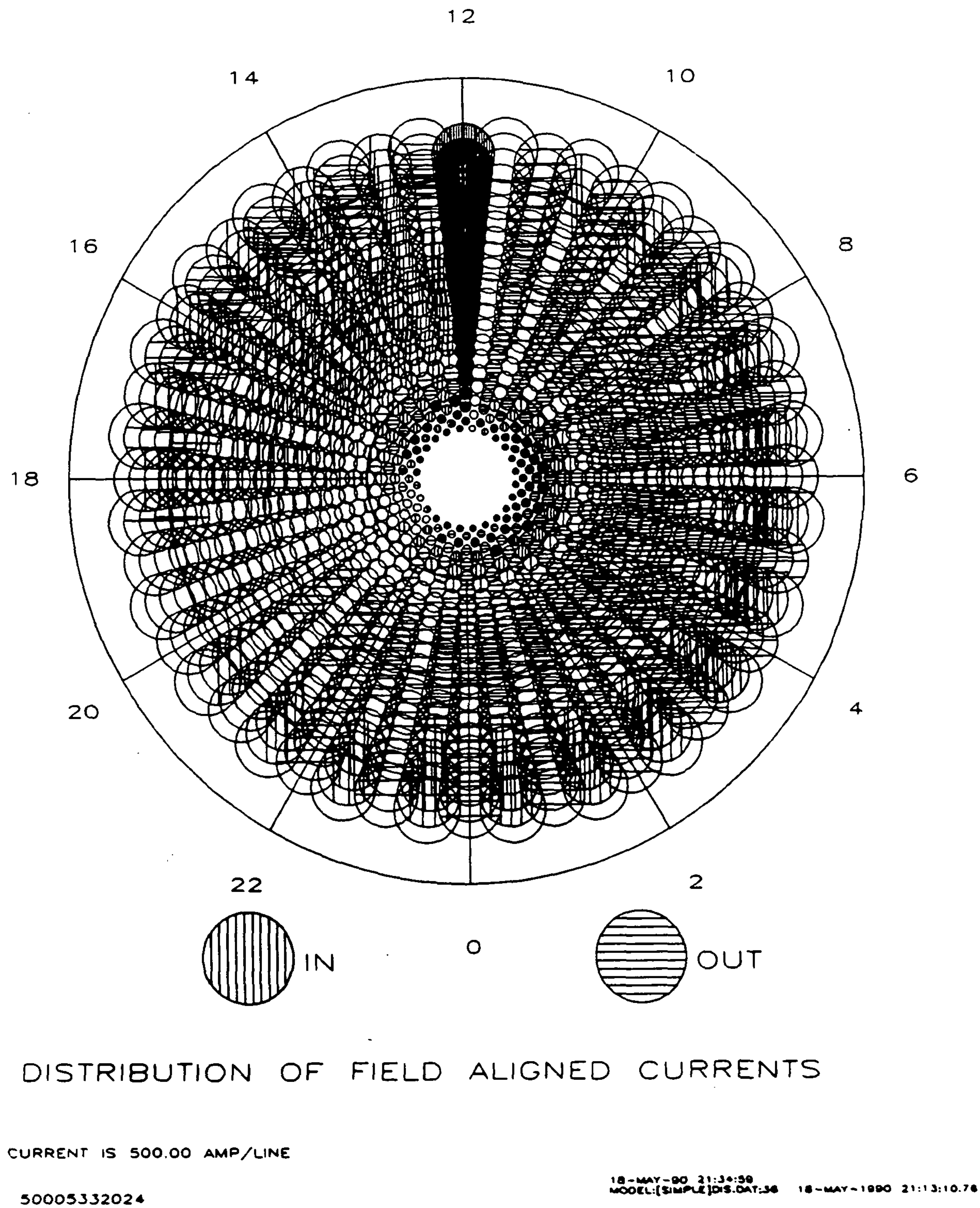

12
14
10
16
8
18
6
20
4
22
2
0
IN
OUT
DISTRIBUTION OF FIELD ALIGNED CURRENTS
CURRENT IS 500.00 AMP/LINE
50005332024
18-MAY-90 21:34:59
MODEL:[SIMPLE]DIS.DAT;36 18-MAY-1990 21:13:10.76

FIGURE 3

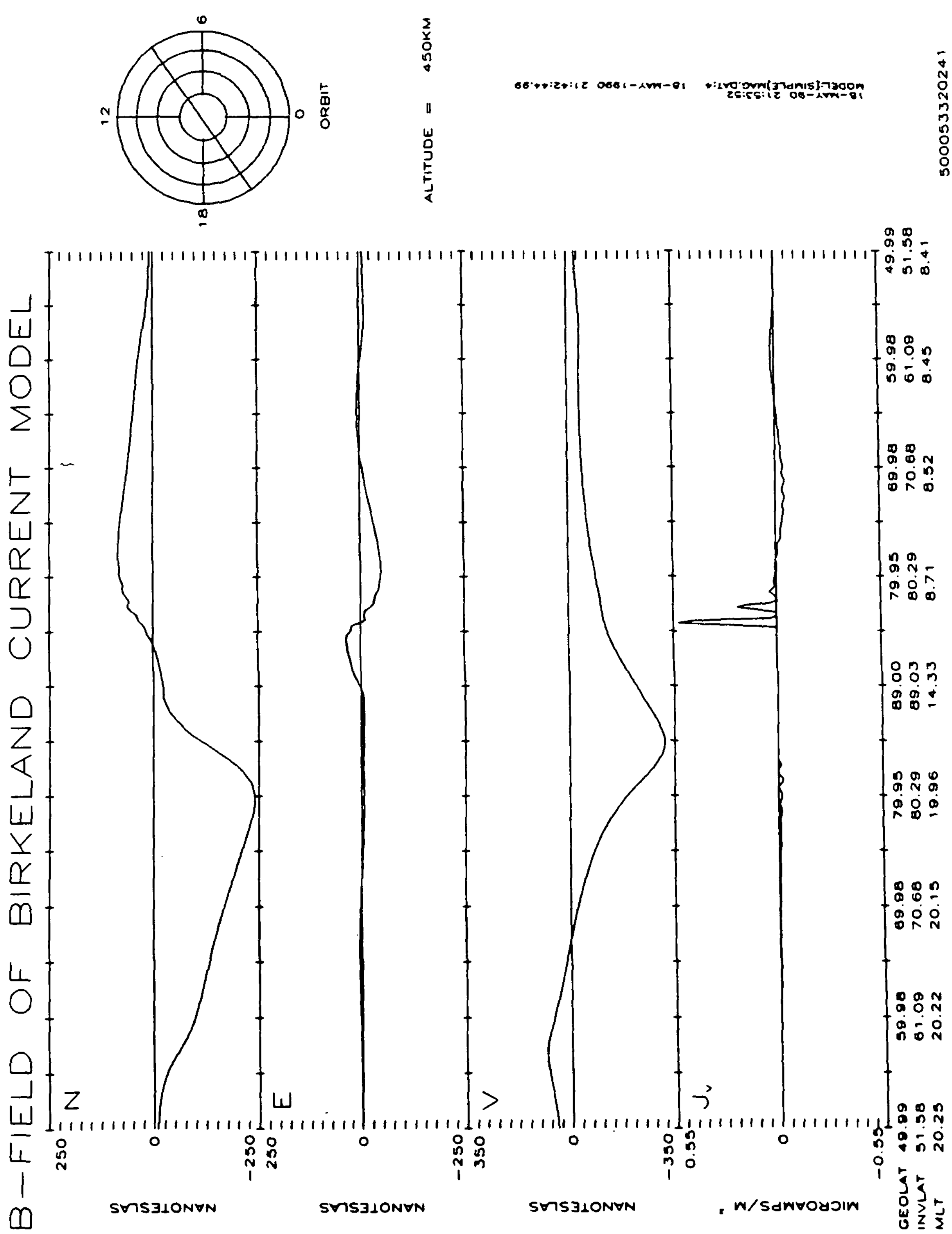

B-FIELD OF BIRKELAND CURRENT MODEL
N
E
V
NANOTESLAS
MICROAMPS/M²
250
0
-250
350
-350
0.55
-0.55
GEOLAT 49.99 59.98 69.98 79.95 89.00 79.95 69.98 59.98 49.99
INVLAT 51.58 61.09 70.68 80.29 89.03 80.29 70.68 61.09 51.58
MLT 20.25 20.22 20.15 19.96 14.33 8.71 8.52 8.45 8.41
12
6
18
0
ORBIT
ALTITUDE = 450KM
500053320241

FIGURE 4

ATTACHMENT A
Abstract of paper submitted to Spring 1990
American Geophysical Union Meeting

A Method for Computing Magnetic Perturbations at Satellite Altitude due to Distributed Currents in the Ionosphere and Magnetosphere

D. M. Klumpar (Lockheed Palo Alto Research Laboratory, Palo Alto, CA, 94304; 415-424-3288; SPAN mail LOCKHD::KLUMP)

Low altitude magnetic survey satellites respond to the main magnetic field, to crustal anomaly fields, and to fields produced by currents in the ionosphere-magnetosphere system. One sub-discipline of space sciences endeavors to describe the space currents and relate them to processes in the magnetosphere and ionosphere. It is generally assumed that the entire observed magnetic perturbation is due to currents close to the satellite with a simple planar sheet current geometry. Given these assumptions one obtains an approximation for the local field-aligned currents. Contributions from distant sources are ignored. In the field of terrestrial geomagnetism the space currents represent a large temporally varying source of contaminating signal that must be removed to deduce the main field and/or crustal anomaly fields. Steep gradients due to local effects are readily recognized but larger-scale perturbations are a severe cause of inaccuracies. We illustrate here the use of a modeling routine, which has application to each of the above disciplines. The technique computes the magnetic field along a satellite orbit due to distributed electrical currents in the ionosphere and magnetosphere. It takes as input a description of the distributed currents over the entire high latitude ionosphere and computes the magnetic contribution at each point from the ensemble of currents in the system. Using this technique we input realistic current distributions and calculate the resulting magnetic perturbations along a satellite orbit. The modeling software is useful as a research tool for analyzing the relative contributions of local and more distant currents at a satellite and therefore generally as a tool in understanding the sources of the magnetic purtabation signatures. Various portions of the input currents can be controlled independently, or turned off completely, so as to allow a versatility to "experiment" that is not found in nature. Since the model requires an ad hoc input current system it is not, in its present form, suitable for uniquely determining the currents that give rise to a specific measured magnetic perturbation profile. For that one requires a solution to the inverse problem. As an application of the technique we show distributed current systems which produce magnetic signatures that compare favourably with actual Magsat measurements and illustrate how the different components of the current system individually contribute to the magnetic perturbation.

1. 1990 Spring Meeting
2. 000895585
3. (a) D. M. Klumpar
   Dept 91-20/B255
   Lockheed
   3251 Hanover Street
   Palo Alto, CA 94304

   (b) 415-424-3288
4. GP
5. (a) GP08 Magnetic Disturbances: Separation, Modeling and Applications

   (b) 2708 Current Systems, 1594 Instr and Techniques
6.
7. 0%
8. Invoice $60 to PO# ZAP22568 (Copy attached)
   Attn: Lois Fulbright
   Technical Information
   Dept 90-11, Bldg 201
   3251 Hanover Street
   Palo Alto, CA 94304
   Phone 415-424-2810
9. C
10. none
11. No

ATTACHMENT B
FORTRAN Source Codes
Magnetic Field Model

SCURDIS

SOURCE CODE

```
c  SCURDIS.FOR  revision history, Summer, 1988
c               John L. Jamison, for Dr. DM Klumpar
c               Lockheed Palo Alto Research Laboratory
c               O/91-20 B/255
c
c   6/29/88     JLJ     made this "Simple" version of curdis
c                       added INCLUDE for standardization of data decl.
c                       changed AMPS to amp
c
c   6/28/88     JLJ     added "Units" in value input prompts, and
c                       input-verification code
c   6/28/88     JLJ     changed E-W currents in loop 40000
c
C*******************************************************************)
C
C **** CURDIS DEFINES A CURRENT DISTRIBUTION ARRAY AND
C **** PUTS IT INTO A DATA FILE, DIS.DAT
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C

      subroutine SCURDIS(inter,pnumt,pnuml,pol1,pol2,prf,pdf,pncode)

        include 'model:[simple]abrk.inc'

      DIMENSION CL(2,21)
      DIMENSION TCL(2,21)
      DIMENSION CLI(2,21), TMU(2,21)

      REAL*4 LAMBDA
      DATA PI / 3.14159265 /
      DATA RE / 6371000. /
      DATA ALTI / 120000. /
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      integer pnumt,pnuml,pncode
      real prf,pdf,pol1,pol2

      if (inter.eq.0) then
        numl=pnuml
        numt=pnumt
        rf=prf
        df=pdf
        ol1=pol1
        ol2=pol2
        ncode=pncode
        goto 2000
      endif

      write (5,*)
      write (5,*) '    CURDIS.EXE'
      write (5,*)
00999 WRITE (5,01000)
01000 FORMAT ('$Enter number of Cell Rings [ INTEGER,1..20 ] : ')
      READ (5,*) NUMT
      if (numt.lt.1.or.numt.gt.20) goto 999
C

01009 WRITE (5,01010)
01010 FORMAT ('$Enter number of cells per 360 deg longitude '
     *        '[ INTEGER, 1..24 ] : ')
      READ (5,*) NUML
      if (numl.lt.1.or.numl.gt.24) goto 1009

C
```

```
01019 WRITE (5,01020)
01020 FORMAT('$Enter inner and outer colatitudes of ring '
     *        '[ REAL, 0.0 .. 90.0 ] : ')
      READ (5,*) CL1, CL2
      if ((cl1.lt.0.0.or.cl1.gt.90.0).or.(cl2.lt.0.0.or.cl2.gt.90.0))
     *  goto 1019

C
      WRITE (5,01040)
01040   FORMAT('$Enter max radius of current filaments ',
     *        ' [ REAL, meters ] : ')
      read (5,*) rf

      WRITE (5,01050)
01050 FORMAT('$Enter latitudinal thickening exponent : ')
      READ (5,*) DF
C
      WRITE (5,01060)
01060 FORMAT ( '$Enter model number : ' )
      READ (5,*) NCODE


 2000 continue
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOP 10000 DEFINES THICKNESS OF CURRENT FILAMENTS:
C ****  FOR F-A CURRENTS SUPPLYING E-W CURRENTS ON I = 1;
C ****  FOR N-S CURRENTS ON I = 2;
C ****  FOR E-W CURRENTS ON I = 3.
C
      DO 10000 I = 2, 4
         DO 10000 J = 1, NUMT
            DO 10000 K = 1, NUML
               TF(I,J,K) = 1.089/(((((J-.5)*(CL2-CL1)
     1                     /NUMT + CL1)/CL2)**DF)*(RF**2))
10000  CONTINUE
C
C **** LOOP 20000 DEFINES THICKNESS OF CURRENT FILAMENTS
C **** FOR F-A CURRENTS SUPPLYING N-S CURRENTS.
C
      N = NUMT + 1
      DO 20000 J = 1, N
         DO 20000 K = 1, NUML
            TF(1,J,K) = 1.089/(((((J-1)*(CL2-CL1)/NUMT + CL1)
     1                  /CL2)**DF)*(RF**2))
20000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOPS 30000 AND 40000 DEFINE CURRENT PER LOOP FOR N-S AND E-W RESP.
C
      DO 30000 J = 1,NUMT
         DO 30000 K = 1, NUML
            S = -1./2.
            IF (K.GT.NUML/2.) S = 1./2.
            amp(1,J,K) = S
30000 CONTINUE
      DO 40000 J = 1,NUMT
         DO 40000 K = 1, NUML
            S=-.5
            IF (K .GT. 12.*NUML/24.) S = 0.5
            amp(2,J,K) = S
40000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
```

```
C
C **** LOOPS 50000 AND 60000 DEFINE LENGTH OF CURRENT FILAMENTS
C **** AND THEIR LOCATION IN SPACE
C
      DO 50000 I = 1, 2
         DO 50000 J = 1, N
            CL(I,J) = ((CL2-CL1)*(I*J-1)/(I*NUMT) + CL1)*PI/180.
            SCL(I,J) = SIN(CL(I,J))
            CCL(I,J) = COS(CL(I,J))
            TCL(I,J) = TAN(CL(I,J))
            CLI(I,J) = ATAN(2./TCL(I,J))
            CMU(I,J) = SIN(CLI(I,J) - CL(I,J))
            SMU(I,J) = COS(CLI(I,J) - CL(I,J))
            TMU(I,J) = 1./TAN(CLI(I,J) - CL(I,J))
            REF(I,J) = (RE + ALTI)*(CCL(I,J) - SCL(I,J)/TMU(I,J))
            RB(I,J) = (RE + ALTI)*SCL(I,J)/SMU(I,J)
            RT(I,J) = (RE + ALTI)*3.
50000 CONTINUE
C
      DO 60000 K = 1, NUMT
         RSI(K) = (RE + ALTI)*(COS(CL(1,K)-CL(2,K)))/CCL(2,K)
         RI(1,K) = RSI(K)*SCL(2,K)
     1             - (RE + ALTI)*SIN(CL(2,K)- CL(1,K))
         RI(2,K) = RI(1,K) + 2.*(RE + ALTI)
     1             *SIN(CL(2,K) - CL(1,K))
         RZR(K) = RE + ALTI
         REJ(K) = RZR(K)*TAN(PI/NUML)
60000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOP 70000 DEFINES THE RING CURRENT
C
      RINGA=4.*RE
      RINGB=4.*RE*TAN(PI/NUML)
      DO 70000 I=1,NUML
         TPR(I)=1.089/(1000**2)
         AMPR(I)=1.
70000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      OPEN (UNIT=1,NAME='DIS.DAT',TYPE='NEW')
      WRITE (1,*) NCODE, DF
      WRITE (1,*) CL1, CL2
      WRITE (1,*) NUMT, NUML
      WRITE (1,*) (((TP(I,J,K), K=1,NUML), J=1,N), I=1,4)
      WRITE (1,*) (((amp(I,J,K), K=1,NUML), J=1,NUMT), I=1,2)
      WRITE (1,*) ((REF(I,J), J=1,N), I=1,2)
      WRITE (1,*) ((RB(I,J), J=1,N), I=1,2)
      WRITE (1,*) ((RT(I,J), J=1,N), I=1,2)
      WRITE (1,*) (RSI(K), K=1,NUMT)
      WRITE (1,*) (RZR(I), I=1,NUMT)
      WRITE (1,*) (REJ(I), I=1,NUMT)
      WRITE (1,*) ((RI(I,J), J=1,NUMT), I=1,2)
      WRITE (1,*) ((SCL(I,J), J=1,N), I=1,2)
      WRITE (1,*) ((CCL(I,J), J=1,N), I=1,2)
      WRITE (1,*) ((SMU(I,J), J=1,N), I=1,2)
      WRITE (1,*) ((CMU(I,J), J=1,N), I=1,2)
      WRITE (1,*) RINGA, RINGB
      WRITE (1,*) (TPR(I),I=1,NUML)
      WRITE (1,*) (AMPR(I),I=1,NUML)
      CLOSE (UNIT=1)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
```

```
RETURN
END

program scurdis_it
  implicit none
  call scurdis(1,......)
end
```

SAMPLT

SOURCE CODE

```
c   SAMPLT              Birkfield Current Ampere Plot
c
c                        JL Jamison, for DM Klumpar
c                        Lockheed Palo Alto Research Laboratory
c                        O/91-20  B/255
c
c   Revision History
c
c        2-Jul-88        JLJ     cleaned up input prompting
c       30-Jun-88        JLJ     added SBRK.INC to standardize declarations
c
c
C**********************************************************************)
C
C **** AMPLT SHOWS CURRENT FLOWING THROUGH THE SURFACE OF A SPHERE
C **** JUST ABOVE THE IONOSPHERIC CURRENTS.
C **** EACH CIRCLE REPRESENTS A FIELD ALIGNED CURRENT FILAMENT AND
C **** ABOUT 90 PERCENT OF THE PLATICURTICALLY DISTRIBUTED  CURRENT
C **** IS THEREIN ENCLOSED.
C **** EACH LINE IN ONE OF THESE CIRCLES REPRESENTS ONE AMP (NUMBER
C **** OF LINES + OR - .5 IS CURRENT).
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      subroutine samplt(inter,pfnum,pdiv)

        include 'model:[simple]sbrk.inc'   ! JLJ

c       DIMENSION AMP(2,20,24), TP(4,21,24)

      character *2 fnum,pfnum
      character *11 darray  /'DIS.DAT;  '/
      integer inter
      real pdiv

      REAL*4 INCL, MLT
      DATA PI / 3.14159265 /
      DATA CF / 1.2594E-6 /
      DATA INCL, THTA / 'INCL', 'THTA' /
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      write (6,997)
00997   format('4')
      write (6,*) '    SAMPLT.EXE       Birkeland Field-Aligned '
     * //'Current Distribution'
      write (6,998)
00998   format('3')

      if (inter.eq.0) then
         fnum(1:2)=pfnum(1:2)
         div = pdiv
         goto 2000
      endif

      WRITE (6,01000)
01000 FORMAT ('$Enter DIS.DAT version number [ Integer ] : ')
      READ (6,01010) FNUM
01010 FORMAT (2A)
C
      WRITE (6,01020)
01020 FORMAT ('$Enter circlegram DIV factor [ Real ] : ')
      READ (6,*) DIV
C
02000 DARRAY(9:10)=FNUM(1:2)
```

```
      if (inter.eq.1) then
        write (6,*) 'File : ',darray(1:10)
        write (6,*) 'Div  : ',div
        write (6,*)
      endif

      CALL CALCMP(X,Y,2,0)
      CALL CALCMP(X,Y,0,2)
      XORG = 5.6
      YORG = 5.5
      CALL CALCMP(XORG,YORG,0,3)
      CALL GRAIN(0.0)

      call putinfo(darray(1:11),7.8,1.5,.1,.false.)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      OPEN (UNIT=1,NAME=DARRAY,TYPE='OLD')
      READ (1,*) NCODE
      READ (1,*) CL1, CL2
      READ (1,*) NUMT, NUML
      N = NUMT + 1
      READ (1,*) (((TP(I,J,K), K=1,NUML), J=1,N), I=1,4)
      READ (1,*) (((AMP(I,J,K), K=1,NUML), J=1,NUMT), I=1,2)
      CLOSE (UNIT=1)
C
C
      N = NUMT + 1
      DO 20000 NT = 1, 2
         DO 20000 I = 1, N
            DO 20000 J = 1, NUML
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::)
C
C **** THIS SECTION CALCULATES THE CURRENT PER FILAMENT
C
               IF((NT .EQ. 2) .AND. (I .EQ. N)) GO TO 20000
               F = ((I-1./NT)*(CL2-CL1)/NUMT + CL1)/40.
               XL = -4.5*F*COS(2.*(J+NT/2.-1.)*PI/NUML)
               YL = -4.5*F*SIN(2.*(J+NT/2.-1.)*PI/NUML)
               FAMP = AMP(1,1,J)
               IF((NT .EQ. 2) .AND. (J .LT. NUML))
     1         FAMP = AMP(2,I,J+1) - AMP(2,I,J)
               IF((NT .EQ. 2) .AND. (J .EQ. NUML))
     1         FAMP = AMP(2,I,1) - AMP(2,I,NUML)
               IF((NT .EQ. 1) .AND. (I .GT.1))
     1         FAMP = AMP(1,I,J) - AMP(1,I-1,J)
               IF((NT .EQ. 1) .AND. (I .EQ. N)) FAMP = -AMP(1,NUMT,J)
               IF(FAMP .EQ. 0.) GO TO 20000
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::)
C
C **** LOOP 21000 CALCULATES THE RADIUS OF EACH FILAMENT
C **** AND DRAWS THE CIRCLE REPRESENTING IT.
C
               RF = SQRT(1.089/TP(NT,I,J))
      CALL CALCMP (XL,YL,0,1)
      PRAD=RF*CF
      CALL CALCMP (XL,YL,1,-5)
      CALL ARC (PRAD,0.,360.)
      CALL CALCMP (XL,YL,0,-5)
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::)
C
C **** LOOPS 22000 AND 23000 DRAW THE LINES IN EACH CIRCLE THAT
```

```
C **** SHOW THE CURRENT IN EACH FILAMENT, 22000 FOR IN AND 23000 FOR OUT.
C
               IF(ABS(FAMP/DIV) .LT. .05) GO TO 20000
               NLINE = INT(ABS(FAMP*10./DIV) + .5)
               IF(FAMP .LT. 0.) GO TO 20300
               DO 22000 L = 1, NLINE
                  XW = RF*CF*SQRT(1.-(1.-L/((NLINE+1)/2.))**2)
                  X = XL - XW
                  Y = YL + RF*CF*(1.-L/((NLINE+1)/2.))
                  CALL CALCMP(X,Y,0,1)
                  X = XL + XW
                  Y = YL + RF*CF*(1.-L/((NLINE+1)/2.))
                  CALL CALCMP(X,Y,1,1)
22000          CONTINUE
               IF(FAMP .GT. 0.) GO TO 20000
20300          DO 23000 M = 1, NLINE
                  YW = RF*CF*SQRT(1.-(1.-M/((NLINE+1)/2.))**2)
                  X = XL + RF*CF*(1.-M/((NLINE+1)/2.))
                  Y = YL - YW
                  CALL CALCMP(X,Y,0,1)
                  X = XL + RF*CF*(1.-M/((NLINE+1)/2.))
                  Y = YL + YW
                  CALL CALCMP(X,Y,1,1)
23000          CONTINUE
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::)
C
20000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOPS 30000 AND 40000 DRAW THE LATITUDE CIRCLES AND
C **** MLT LINES AND LABELS RESPECTIVELY.
C
      CALL CALCMP (0.,0.,0,1)
      DO 30000 I = 1, 4
        PRAD=1.125*I
        CALL ARC (PRAD,0.,360.)
30000 CONTINUE
C
      DO 40000 I = 1, 12
         TH = (I-1)*PI/6.
         ST = SIN(TH)
         CT = COS(TH)
         XL = 4.5*CT
         YL = 4.5*ST
         X = XL - 3.375*CT
         Y = YL - 3.375*ST
         CALL CALCMP(X,Y,0,1)
         X = XL
         Y = YL
         CALL CALCMP(X,Y,1,1)
         MLT = 2.*(I-1)
         XN = XL + .7*CT + .105
         YN = YL + .5*ST - .07
         IF (MLT .GT.9) YN = YN - .14
         CALL NUMBER(XN,YN,.21,MLT,90.,-1)
40000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOP 50000 DRAWS TWO REPRESENTATIVE FILAMENT CROSS-SECTIONS
C **** ONE SHOWING CURRENT IN, THE OTHER, CURRENT OUT, EACH HAS
C **** A RADIUS OF 400000 METERS AND A CURRENT OF TEN AMPS.
C
      ROP = 400000.
```

```
      DO 50000 I = 1, 2
         GO TO (50010,50020) I
50010    XL = 5.3
         YL = -2.5
         GO TO 50030
50020    XL = 5.3
         YL = 1.9
50030    PRAD=ROP*CF
         CALL CALCMP (XL,YL,0,1)
         CALL ARC (PRAD,0.,360.)
         GO TO (50040,50050) I
50040    DO 52000 J = 1, 10
            XW = ROP*CF*SQRT(1.-(1.-J/5.5)**2)
            X = XL - XW
            Y = YL + ROP*CF*(1.-J/5.5)
            CALL CALCMP(X,Y,0,1)
            X = XL + XW
            Y = YL + ROP*CF*(1.-J/5.5)
            CALL CALCMP(X,Y,1,1)
52000    CONTINUE
         GO TO 50000
50050    DO 53000 J = 1, 10
            YW = ROP*CF*SQRT(1.-(1.-J/5.5)**2)
            X = XL + ROP*CF*(1.-J/5.5)
            Y = YL - YW
            CALL CALCMP(X,Y,0,1)
            X = XL + ROP*CF*(1.-J/5.5)
            Y = YL + YW
            CALL CALCMP(X,Y,1,1)
53000    CONTINUE
50000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      CALL SYMBOL(6.7,-5.07,.28,22HDISTRIBUTION OF FIELD ,90.,22)
      CALL SYMBOL(999.,999.,.28,16HALIGNED CURRENTS,90.,16)
      CURR=DIV/10.
      CALL SYMBOL(7.5,-5.21,.14,11HCURRENT IS ,90.,11)
      CALL NUMBER(999.,999.,.14,CURR,90.,2)
      CALL SYMBOL(999.,999.,.14,9H AMP/LINE,90.,9)
      CALL SYMBOL(5.5,-1.9,.28,2HIN,90.,2)
      CALL SYMBOL(5.5,2.5,.28,3HOUT,90.,3)
      RCODE = NCODE
      CALL NUMBER(8.0,-5.07,.14,RCODE,90.,0)
      CALL NUMBER(999.,999.,.14,CL1,90.,0)
      CALL NUMBER(999.,999.,.14,CL2,90.,0)
      RUMT=NUMT
      CALL NUMBER(999.,999.,.14,RUMT,90.,0)
      RUML=NUML
      CALL NUMBER(999.,999.,.14,RUML,90.,0)

      CALL PAUS
      CALL CALCMP(X,Y,1000,2)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      RETURN
      END

      program samplt_it
        call samplt (1,'  ',0.0)
        stop
      end
```

SCURPLT

SOURCE CODE

```
c    SCURPLT.FOR              Birkland Current Distribution Plotter
c
c                          JL Jamison,    for DM Klumpar
c                          Lockheed Palo Alto Research Laboratory
c                          O/91-20   B/255
c
c       Revision History
c
c        12-Jul-88       JLJ     made into callable procedure
c         2-Jul-88       JLJ     cleaned up input prompts
c        30-Jun-88       JLJ     added 'SBRK.INC' to standardize declarations
c
c
C***********************************************************************)
C
C **** CURPLT SHOWS THE CURRENT VECTORS IN THE IONOSPHERE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      subroutine scurplt(inter,pfnum,pcur)
        include 'model:[simple]sbrk.inc'
      character*11 darray /'DIS.DAT;  '/
      character*2 fnum,pfnum
      integer*4 inter

      REAL*4 MLT, pcur
      DATA PI / 3.14159265 /
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C

      write (6,997)
0997    format('4')
      write (6,*) '      SCURPLT.EXE     Birkeland Current Distribution'
     *  // ' Plotter'
      write (6,*) '                                   Simple Model'
      write (6,998)
0998    format('2')

      if (inter.eq.0) then
        fnum(1:2)=pfnum(1:2)
        cur=pcur
        goto 2000
      endif

      WRITE (6,01000)
01000 FORMAT ('$Enter DIS.DAT version number [ integer ] : ')
      READ (6,01010) FNUM
01010 FORMAT (2A)
C
      WRITE (6,01020)
01020 FORMAT ( '$Enter Current Magnitude [ Real ] : ' )
      READ (6,*) CUR
C

02000 DARRAY(9:10)=FNUM(1:2)

      if (inter.eq.1) then
        write (6,*) 'File    : ',darray(1:10)
        write (6,*) 'Current : ',cur
        write (6,*)
      endif

      OPEN (UNIT=1,NAME=DARRAY,TYPE='OLD')
      READ (1,*) NCODE,DF
      READ (1,*) CL1, CL2
```

```
      READ (1,*) NUMT, NUML
      READ (1,*) (((TP(I,J,K), K=1,NUML), J=1,NUMT+1), I=1,4)
      READ (1,*) (((AMP(I,J,K), K=1,NUML), J=1,NUMT), I=1,2)
      CLOSE (UNIT=1)
C
      CALL CALCMP (X,Y,1,0)
      CALL CALCMP (X,Y,0,2)
      CALL WINDEW (0.,14.,-5.6,8.4,0.,11.,-5.5,5.5,1)

        call putinfo(darray(1:11),7.8,2.3,.10,.false.)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** LOOP 10000 DRAWS THE VECTORS AND THEIR HEADS ****
C
      DO 10000 I = 1, NUMT
         DO 10000 J = 1, NUML
C
C **** THIS PART DRAWS THE VECTORS. ****
           F=(CL1+(2.*I-1.)*(CL2-CL1)/(2.*NUMT))/40.
           XL=-4.5*F*COS(2.*(J-.5)*PI/NUML)
           YL=-4.5*F*SIN(2.*(J-.5)*PI/NUML)
           CALL CALCMP (XL,YL,0,1)
           IF ((AMP(1,I,J)**2+AMP(2,I,J)**2).EQ.0.) GO TO 10000
           XF=XL+CUR*(AMP(1,I,J)*XL-AMP(2,I,J)*YL)/SQRT(XL**2+YL**2)
           YF=YL+CUR*(AMP(1,I,J)*YL+AMP(2,I,J)*XL)/SQRT(XL**2+YL**2)
           CALL CALCMP (XF,YF,1,1)
C
C **** THIS PART DRAWS THE ARROW HEADS. ****
           TH=ATAN2((XF-XL),(YL-YF))
           X=XF-.06*SIN(TH-20.*PI/180.)
           Y=YF+.06*COS(TH-20.*PI/180.)
           CALL CALCMP (X,Y,1,1)
           X=XF-.06*SIN(TH+20.*PI/180.)
           Y=YF+.06*COS(TH+20.*PI/180.)
           CALL CALCMP (X,Y,1,1)
           CALL CALCMP (XF,YF,1,1)
C
10000 CONTINUE
C
C **** LOOPS 50000 AND 60000 DRAW THE LATITUDE CIRCLES ****
C **** AND MLT LINES AND LABELS RESPECTIVELY.          ****
      CALL CALCMP (0.,0.,0,1)
      CALL GRAIN (0.0)
      DO 50000 I = 1, 4
         PRAD=1.125*I
         CALL ARC (PRAD,0.,360.)
50000 CONTINUE
C
      DO 60000 I = 1, 12
         ST=SIN((I-1)*PI/6.)
         CT=COS((I-1)*PI/6.)
         XL=4.5*CT
         YL=4.5*ST
         X=XL-3.375*CT
         Y=YL-3.375*ST
         CALL CALCMP (X,Y,0,1)
         CALL CALCMP (XL,YL,1,1)
         MLT=2.*(I-1)
         XN=XL+.7*CT+.105
         YN=YL+.7*ST-.07
         IF (MLT.GT.9) YN=YN-.14
         CALL NUMBER (XN,YN,.21,MLT,90.,-1)
60000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
```

```
C
      CALL CALCMP (7.5,-4.86,0,1)
      CALL CALCMP (7.0,-4.86,1,1)
      CALL CALCMP (7.05,-4.88,1,1)
      CALL CALCMP (7.05,-4.84,1,1)
      CALL CALCMP (7.0,-4.86,1,1)
      CURR=.5/CUR
      CALL SYMBOL (7.35,-4.7,.14,2H= ,90.,2)
      CALL NUMBER (999.,999.,.14,CURR,90.,2)
      CALL SYMBOL (999.,999.,.14,4H AMP,90.,4)
      CALL SYMBOL (6.4,-5.07,.28,16HDISTRIBUTION OF ,90.,16)
      CALL SYMBOL (999.,999.,.28,20HIONOSPHERIC CURRENTS,90.,20)
      RCODE=NCODE
      CALL NUMBER (8.0,-5.07,.14,RCODE,90.,0)
      CALL NUMBER (999.,999.,.14,CL1,90.,0)
      CALL NUMBER (999.,999.,.14,CL2,90.,0)
      RUMT=NUMT
      CALL NUMBER (999.,999.,.14,RUMT,90.,0)
      RUML=NUML
      CALL NUMBER (999.,999.,.14,RUML,90.,0)
      CALL PAUS
      CALL CALCMP (X,Y,1000,2)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      RETURN
      END

      program scurplt_it
        call scurplt(I,'  ',0.0)
        stop
      end
```

SBRKALC

SOURCE CODE

```
c  SBRKALC.FOR                          Birkeland Current calculation
c                          John L. Jamison  for DM Klumpar
c                          Lockheed Palo Alto Research Laboratory  O/91-20 B/255
c                          June, 1988
c
c  Revision history
c
c    12-Jul-88    JLJ      made into callable subroutine
c     6-Jul-88    JLJ      Added max to NMEAS
c     1-Jul-88    JLJ      Debugging
c    30-Jun-88    JLJ      change ALTI from 140000. to 120000.
c    30-Jun-88    JLJ      added input value checking code
c                          fixed prompt for altitude
c    29-Jun-88    JLJ      initial entry
c
c=======================================================================
      subroutine SBRKALC(inter,pfnum,palt,pincl,ptheta,pnmeas,
     *                   pifld,pipass)

      dimension fld(4), ftem(3)

c
c SBRK.INC defines the common block, and most of the shared data types
c
      include 'model:[simple]sbrk.inc'
      integer inter, pnmeas,pifld,pipass
      real palt, pincl,ptheta

      character*11 darray /'DIS.DAT;  '/
      character*2 fnum,pfnum
      real*4 incl,mp
      data RE /6371000./
      data alti /120000./   !JLJ
      data pi /3.14159265/

      write (6,*)
      write (6,*) '  SBRKALC.EXE  (Simple)  Birkeland Current Generator'
      write (6,*)

      if (inter.eq.0) then
         fnum(1:2)=pfnum(1:2)
         alt=palt
         ipass= pipass
         ifld = pifld
         incl = pincl
         theta= ptheta
         nmeas= pnmeas
         goto 2000
      endif

      write (6,1000)
 1000   format('$Enter DIS.DAT version number: ')
      read (5, 1010)
 1010   format(2a)

      write (6,1020)
 1020   format('$Enter Altitude [ REAL, meters ] : ')
      read (5,*) alt
c
c  if inputted ALT is less than 10000, then its assumed that the
c  user entered a KM value
c
      if (alt.gt.0.0.and.alt.lt.10000.) then
         write(6,*) ' Assuming Kilometers, adjusting...'
         alt=alt*1000.
         write(6,1019) alt
```

```
1019        format(' Using ',f8.0,' meters for altitude.')
      endif

      write (6, 1021)
1021    format('$Enter Inclination [ REAL, deg. from pole ] : ')
      read (5,*) incl

      write (6, 1022)
1022    format('$Enter Theta [ REAL, deg from dawn-dusk line ] : ')
      read (5,*) theta

1029  write (6, 1030)
1030  format('$Enter number of measurement points [ Integer, '
     *   ,'<= 500 ] : ')
      read (5,*) nmeas
      if (nmeas.gt.500) goto 1029

1039  write (6, 1040)
1040    format(' Enter 1 for the Field of all currents'/
     *          '       2 for Field-Aligned only'/
     *          '       3 for North-South only'/
     *          '       4 for East-West only'/
     *          '       5 for Ring current only'/
     *          '$> ')
      read (5,*) ifld
      if (ifld.lt.1.and.iflt.gt.5) goto 1039

1049  write (6, 1050)
1050    format(' Enter 1 for Polar'/
     *          '       2 for Equatorial West'/
     *          '       3 for Equatorial East'/
     *          '$> ')
      read (5,*) ipass
      if (ipass.lt.0.and.ipass.gt.3) goto 1049
      if (ipass.eq.0) ipass=1


c  NOTE use of funD for Degrees, rather than fun for Radians   JLJ
 2000 darray(9:10) = fnum(1:2)

      if (inter.eq.1) then
         write(6,*) 'File   : ',darray(1:10)
         write(6,*) 'Alt    : ',alt
         write(6,*) 'Ipass  : ',ipass
         write(6,*) 'Ifld   : ',ifld
         write(6,*) 'Incl   : ',incl
         write(6,*) 'Theta  : ',theta
         write(6,*) 'Nmeas  : ',nmeas
         write(6,*)
      endif

      st= sinD(theta)
      ct= cosD(theta)
      sincl = sinD(incl)
      cincl = cosD(incl)

      open(unit=1,name=darray,type='old')

c
c  following read statements match CURDIS.FOR (Simple)  WRITEs  JLJ
c
      read (1,*) ncode, df
      read (1,*) cl1, cl2
      read (1,*) numt, numl

      n = numt + 1
```

```
      read (1,*) (((tp(i,j,k), k=1,numl), j=1,n), i=1,4)
      read (1,*) (((amp(i,j,k), k=1,numl), j=1,numt), i=1,2)
      read (1,*) ((rzf(i,j), j=1,n), i=1,2)
      read (1,*) ((rb(i,j), j=1,n), i=1,2)
      read (1,*) ((rt(i,j), j=1,n), i=1,2)
      read (1,*) (rzi(k), k=1,numt)
      read (1,*) (rze(i), i=1,numt)
      read (1,*) (rej(i), i=1,numt)
      read (1,*) ((ri(i,j), j=1,numt), i=1,2)
      read (1,*) ((scl(i,j), j=1,n), i=1,2)
      read (1,*) ((ccl(i,j), j=1,n), i=1,2)
      read (1,*) ((smu(i,j), j=1,n), i=1,2)
      read (1,*) ((cmu(i,j), j=1,n), i=1,2)
      read (1,*) ringa, ringb
      read (1,*) (tpr(i),i=1,numl)
      read (1,*) (ampr(i),i=1,numl)
      close (unit=1)

      open(unit=1,name='mag.dat',type='new')
      write (1,*) ncode,alt
      write (1,*) incl,theta
      write (1,*) cl1,cl2
      write (1,*) numt,numl
      write (1,*) nmeas,ifld
      write (1,*) ipass
c
c  loop 1000 determines (X,Y,Z) coords of each measurement point and calls
c   MAGMOD to obtain Field parameters for each point
c
      do 10000 lx=1,nmeas
              mp = (1.-(lx-1)*2./(nmeas-1))*pi*40./180.
     *             -(ipass-1)*270.*pi/180.
              smp = sin(mp)
              cmp = cos(mp)
              x1=(re+alt)*(-st*smp+ct*cmp*sincl)
              y1=(re+alt)*(ct*smp+st*cmp*sincl)
              z1=(re+alt)*cmp*cincl
c
c  x1,y1,z1  = (X,Y,Z) of measurement point
c  fld       = the three components of dB and FAC determined from MAGMOD
c  ifld      = tells magmod which current systems to include when crunching away

              call magmod(x1,y1,z1,fld,ifld)
              write(1,*) fld
10000 continue
      close(unit=1)
      return
      end

      program sbrkalc_it
      call sbrkalc(1,'  ',0.0,0.0,0.0,0,0,0)
      stop
      end
```

SMAGMOD

SOURCE CODE

```
c  SMAGMOD.FOR              Magnetic Field Model
c
c                          John L. Jamison
c                          Lockheed Palo Alto Research Laboratory O/91-20 B/255
c
c      Revision History,
c
c      7/14/88 JLJ     created LAMBDAS lookup table for lambda values.
c      7/1/88  JLJ     debugging
c      6/29/88 JLJ     initial entry
c
c
c
c
c*************************************************************************)
c
      subroutine magmod (xl,yl,zl,fld,ifld)
c
c **** calculates current density and magnetic field components
c **** of birkeland current model defined by curdis.
c
c
c
      include 'model:[simple]sbrk.inc'

      dimension famp(21,24),rm(3,3),fld(4)
      dimension bf(3), btf(3), bti(3), bte(3), btr(3)
      real*4 lambda, mp, jt
      data pi / 3.14159265 /
      real*4 lambdas(24)    !   [1..numl]

c  create lambdas lookup table, because they're always calculated the same way
      do j=1,numl
        lambdas(j)=2.*(j-.5)*pi/numl
      end do
c

      do 10000 i = 1,3
         btf(i) = 0.
         bti(i) = 0.
         bte(i) = 0.
         btr(i) = 0.
10000 continue
      jt = 0.
c
      if (ifld.eq.-2) go to 03000
      if (ifld.gt.2) go to 03000
c
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
c
c **** loop 20000 does the field aligned currents supplying
c **** both the e-w and n-s ionospheric currents.
c **** positive current is vertical.
c
      i=numt+1

      do 20000 m = 1, 2
         if (m.eq.2) i=numt
         do 20000 n = 1, i
            do 20000 j=1,numl
            if (m.eq.2) goto 20010
            famp(1,j)=amp(1,1,j)
            famp(numt+1,j)=-amp(1,numt,j)
            if (n.eq.(numt+1)) goto 20020
            if (n.gt.1) famp(n,j)=amp(1,n,j)-amp(1,n-1,j)
            if (famp(n,j).eq.0.) goto 20000
```

```
            goto 20020

20010       famp(n,1)=amp(2,n,numl)-amp(2,n,1)
            if (j.gt.1) famp(n,j)=amp(2,n,j-1)-amp(2,n,j)
            if (famp(n,j).eq.0) goto 20000

c  rearranged and optimized,  JLJ
c20020      lambda=2.*(j-.5)*pi/numl
c           if (m.eq.2) lambda=2.*j*pi/numl
c
20020       if (m.ne.2) then
                lambda = lambdas(j)
            else
                lambda = 2.*j*pi/numl
            endif

            sla = sin(lambda)
            cla = cos(lambda)
c
            rm(1,1) =  cla*cmu(m,n)
            rm(1,2) =  sla*cmu(m,n)
            rm(1,3) = -smu(m,n)
            rm(2,1) = -sla
            rm(2,2) =  cla
            rm(2,3) =  0.
            rm(3,1) =  cla*smu(m,n)
            rm(3,2) =  sla*smu(m,n)
            rm(3,3) =  cmu(m,n)
c
            xf = xl*rm(1,1) + yl*rm(1,2) + (zl-rzf(m,n))*rm(1,3)
            yf = xl*rm(2,1) + yl*rm(2,2) + (zl-rzf(m,n))*rm(2,3)
            zf = xl*rm(3,1) + yl*rm(3,2) + (zl-rzf(m,n))*rm(3,3)
c
            rcf = xf**2 + yf**2
            rbf = sqrt(rcf + (zf-rb(m,n))**2)
            rtf = sqrt(rcf + (zf-rt(m,n))**2)
c
            bxf = famp(n,j)*(tanh(rcf*tp(m,n,j)))*(yf/rcf)
     1          *((rt(m,n)-zf)/rtf + (zf-rb(m,n))/rbf)
            byf = -famp(n,j)*(tanh(rcf*tp(m,n,j)))*(xf/rcf)
     1          *((rt(m,n)-zf)/rtf + (zf-rb(m,n))/rbf)

            do 21000 ip = 1,3
               bf(ip) = bxf*rm(1,ip) + byf*rm(2,ip)
               btf(ip)= btf(ip) + bf(ip)
21000       continue

            if ((tp(m,n,j)*rcf) .gt. 40.) go to 20000
d           write (6,*) '-----------------------------------'
d           write (6,*) 'M,N,J     =  ',m,n,j
d           write (6,*) 'famp(n,j)=  ',famp(n,j)
d           write (6,*) 'tp(m,n,j)=  ',tp(m,n,j)
d           write (6,*) 'pi        =  ',pi
d           write (6,*) 'rcf       =  ',rcf
d           write (6,*) 'jt        =  ',jt
d           write (6,*) 'cosh      =  ',cosh(tp(m,n,j)*rcf)
            jt = famp(n,j)*tp(m,n,j)
     1          /(pi*((cosh(tp(m,n,j)*rcf))**2)) + jt
20000 continue
c
      if (ifld.eq.2) go to 07000
03000 if (ifld.eq.5) go to 04000
c
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
c
c
```

```
      do 30000 n = 1,numt
         do 30000 j = 1,numl
c             lambda = 2.*(j-.5)*pi/numl
            lambda=lambdas(j)

            sla = sin(lambda)
            cla = cos(lambda)
c
            rm(1,1) =  cla*ccl(2,n)
            rm(1,2) =  sla*ccl(2,n)
            rm(1,3) = -scl(2,n)
            rm(2,1) = -sla
            rm(2,2) =  cla
            rm(2,3) =  0.
            rm(3,1) =  cla*scl(2,n)
            rm(3,2) =  sla*scl(2,n)
            rm(3,3) =  ccl(2,n)
c
            if (ifld.eq.-3) go to 30100
            if (ifld.eq.4) go to 30100
c
c---------------------------------------------------------------------)
c
c **** this part does the north-south ionospheric currents.
c **** negative current is northward.
c
            if(amp(1,n,j) .eq. 0.) go to 30010
c
            xf = xl*rm(1,1) + yl*rm(1,2) + (zl-rzi(n))*rm(1,3)
            yf = xl*rm(2,1) + yl*rm(2,2) + (zl-rzi(n))*rm(2,3)
            zf = xl*rm(3,1) + yl*rm(3,2) + (zl-rzi(n))*rm(3,3)
c
            rcf = yf**2 + zf**2
            ri1 = sqrt(rcf + (xf - ri(1,n))**2)
            ri2 = sqrt(rcf + (xf - ri(2,n))**2)
c
            byf = -amp(1,n,j)*(tanh(100.*rcf*tp(3,n,j)))*(zf/rcf)
     1            *(((xf-ri(1,n))/ri1) + (ri(2,n)-xf)/ri2)
            bzf =  amp(1,n,j)*(tanh(100.*rcf*tp(3,n,j)))*(yf/rcf)
     1            *(((xf-ri(1,n))/ri1) + (ri(2,n)-xf)/ri2)

            do 31000 ip = 1,3
               bf(ip) = byf*rm(2,ip) + bzf*rm(3,ip)
               bti(ip) = bti(ip) + bf(ip)
31000       continue
30010       continue
c
            if (ifld.eq.-4) go to 30000
            if (ifld.eq.3) go to 30000
c
c---------------------------------------------------------------------)
c
c **** this part does the east-west electrojets.
c **** positive current is eastward.
c
30100       if(amp(2,n,j) .eq. 0.) go to 30000
c
            xf = xl*rm(1,1) + yl*rm(1,2) + zl*rm(1,3)
            yf = xl*rm(2,1) + yl*rm(2,2) + zl*rm(2,3)
            zf = xl*rm(3,1) + yl*rm(3,2) + zl*rm(3,3)
c
            rcf = xf**2 + (zf - rze(n))**2
            re1 = sqrt(rcf + (yf + rej(n))**2)
            re2 = sqrt(rcf + (yf - rej(n))**2)
c
            bxf = amp(2,n,j)*(tanh(100.*rcf*tp(4,n,j)))
```

```
     1        *(zf-rze(n))*((yf+rej(n))/rel+(rej(n)-yf)/re2)/rcf
              bzf = -amp(2,n,j)*(tanh(100.*rcf*tp(4,n,j)))
     1        *xf*((yf+rej(n))/rel+(rej(n)-yf)/re2)/rcf
              do 32000 ip = 1,3
                 bf(ip)=bxf*rm(1,ip) + bzf*rm(3,ip)
                 bte(ip) = bte(ip) + bf(ip)
32000         continue
30000 continue
c
      if (ifld.le.1) goto 04000
      if (ifld.lt.5) goto 07000
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
c
04000 if (ifld.eq.-5) goto 07000
c
c **** loop 40000 calculates field due to ring current
      do 40000 i=1,numl

c       lambda=2.*(i-.5)*pi/numl
        lambda=lambdas(i)

        sla=sin(lambda)
        cla=cos(lambda)
c
        rm(1,1)=cla
        rm(1,2)=sla
        rm(1,3)=0.
        rm(2,1)=-sla
        rm(2,2)=cla
        rm(2,3)=0.
        rm(3,1)=0.
        rm(3,2)=0.
        rm(3,3)=1.
c
        if (ampr(i).eq.0.) goto 40000
c
        xf=xl*rm(1,1) + yl*rm(1,2) + zl*rm(1,3)
        yf=xl*rm(2,1) + yl*rm(2,2) + zl*rm(2,3)
        zf=xl*rm(3,1) + yl*rm(3,2) + zl*rm(3,3)
c
        rcf=(xf-ringa)**2 + zf**2
        rr1=sqrt(rcf+(yf+ringb)**2)
        rr2=sqrt(rcf+(yf-ringb)**2)
c
        bxf=ampr(i)*(tanh(rcf*tpr(i)))*zf
     1     *((yf+ringb)/rr1+(ringb-yf)/rr2)/rcf
        bzf=-ampr(i)*(tanh(rcf*tpr(i)))*(xf-ringa)
     1     *((yf+ringb)/rr1+(ringb-yf)/rr2)/rcf
c
        do 41000 ip=1,3
           bf(ip)=bxf*rm(1,ip)+bzf*rm(3,ip)
           btr(ip)=btr(ip)+bf(ip)
41000   continue

40000 continue

c **** loop 70000 adds fields from all sources and converts to s.i.
c
07000 do 70000 ip = 1,3
         fld(ip) = 200.*(btf(ip)+bti(ip)+bte(ip)+btr(ip))
70000 continue
      fld(4) = jt
c
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
c
      return
```

end

SBRKPLT

SOURCE CODE

```
c   SBRKPLT              Bierkland Current density plotter
c
c                        JL Jamison, for DM Klumpar
c                        Lockheed Palo Alto Research Laboratory
c                        O/91-20  B/255
c
c      Revison History
c
c       7/12/88 JLJ      made into callable subroutine
c       7/2/88  JLJ      cleaned up input prompting
c       7/1/88  JLJ      debugging
c       6/30/88 JLJ      Initial integration into "Simple" system
c                        changed ALTI from 140000. to 120000.
c
c
c
C***********************************************************************)
C
C **** BRKPLT PLOTS CURRENT DENSITY AND FIELD COMPONENTS
C **** FOUND BY BRKALC.
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
      subroutine sbrkplt(inter,pfnum,pmode)

      DIMENSION FL(4,500), FTEM(3), FACT(4), FLD(4)
      DIMENSION FMAX(4)
      LOGICAL*1 ATEX(16),TEXT(8)
      character*11 darray  /'MAG.DAT;  '/
      character*2 fnum,pfnum
      integer pmode, inter

      REAL*4 ILAT, INCL, JT, JTMAX, MLT, MP, MAT
      DATA PI / 3.14159265 /
      DATA RE / 6371000. /
      DATA ALTI / 120000. /             !JLJ
      DATA ATEX / 'X', 'Y', 'Z', 'J',
     1            'N', 'E', 'V', 'J',
     2            'S', 'D', 'V', 'J',
     3            'A', 'B', 'Z', 'J' /
C
      write (6,997)
0997    format('4')
      write (6,*) '     SBRKPLT.EXE     Birkeland Current Plotter'
      write (6,*) '                          Simple Model'
      write (6,998)
0998    format('2')

      if (inter.eq.0) then
        fnum(1:2) = pfnum(1:2)
        mode = pmode
        goto 2000
      endif

      WRITE (6,01000)
01000 FORMAT ('$Enter MAG.DAT version number [ Integer ] : ')
      READ (6,01010) FNUM
01010 FORMAT (2A)
C
01015 write (6,*) 'Select Coordinate system:'
      WRITE (6,01020)
01020 FORMAT ('$Enter 1 for XYZ; 2 for NEV; 3 for SDV; '
     *        ,'4 for ABZ [ Integer ] : ')
      READ (6,*) MODE
      if (mode.lt.1.and.mode.gt.4) goto 1015
C
```

```
02000 darray(9:10)=fnum(1:2)

      if (inter.eq.1) then
         write (6,*) 'File : ',darray(1:10)
         write (6,*) 'Mode : ',mode
         write (6,*)
      endif

      OPEN(UNIT=1,NAME=DARRAY,TYPE='OLD')
      READ (1,*) NCODE,ALT
      READ (1,*) INCL, THETA
      READ (1,*) CL1, CL2
      READ (1,*) NUMT, NUML
      READ (1,*) NMEAS,IFLD
      READ (1,*) IPASS
      DO 10000 J = 1,NMEAS
         READ (1,*) FLD
         DO 10000 I = 1,4
            FL(I,J) = FLD(I)
10000 CONTINUE
      CLOSE (UNIT=1)
C
      SINCL = SIN(PI*INCL/180.)
      CINCL = COS(PI*INCL/180.)
      ST = SIN(PI*THETA/180.)
      CT = COS(PI*THETA/180.)
C
      IF (MODE .LE. 1) GO TO 11111
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** CONVERT TO NEV ON 2; SDV ON 3; ABS ON 4
C
      DO 11000 J = 1, NMEAS
         MP = (1.-(J-1)*2./(NMEAS-1))*PI*40./180.
     1        -(IPASS-1)*270.*PI/180.
         SMP = SIN(MP)
         CMP = COS(MP)
         XL = -ST*SMP + CT*CMP*SINCL
         YL =  CT*SMP + ST*CMP*SINCL
         ZL =  CMP*CINCL
         FTEM(1) = FL(1,J)
         FTEM(2) = FL(2,J)
         FTEM(3) = FL(3,J)
         Q1 = SQRT(XL**2 + ZL**2)
         Q2 = SQRT(YL**2 + ZL**2)
         Q3 = SQRT(XL**2 + YL**2)
C
         FL(3,J) = -FTEM(1)*XL - FTEM(2)*YL - FTEM(3)*ZL
C
         IF (MODE .EQ. 3) GO TO 11010
C
         FL(1,J) = -FTEM(1)*XL*ZL/Q3 - FTEM(2)*YL*ZL/Q3 + FTEM(3)*Q3
         FL(2,J) = -FTEM(1)*YL/Q3 + FTEM(2)*XL/Q3
         IF (MODE .EQ.2) GO TO 11000
         FTEM(1) = FL(1,J)
         FTEM(2) = FL(2,J)
         FL(1,J) = FTEM(1)*COS(PI/60.) - FTEM(2)*SIN(PI/60.)
         FL(2,J) = FTEM(1)*SIN(PI/60.) + FTEM(2)*COS(PI/60.)
         GO TO 11000
11010    FL(1,J) = ( FTEM(1)*ZL + FTEM(3)*XL)/Q1
         FL(2,J) = (-FTEM(2)*ZL - FTEM(3)*YL)/Q2
11000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
```

```
C **** FIND MAXIMA
C
11111 DO 12000 I = 1,4
         DO 12000 J = 1,NMEAS
            FMAX(I) = AMAX1(ABS(FL(I,J)),FMAX(I))
12000 CONTINUE
      FMAX(1)=AMAX1(FMAX(1),FMAX(2))
      FMAX(2)=FMAX(1)
C
      CALL CALCMP(X,Y,2,0)
      CALL CALCMP(X,Y,0,2)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** PLOT BACKGROUND
C
      xorg=0.
      yorg=0.
      call calcmp(xorg,yorg,0,3)
      call putinfo(darray(1:10),13.0,1.0,0.10,.false.)

      XORG = .15
      YORG = .1
      CALL CALCMP(XORG,YORG,0,3)
C
      DO 20000 I = 1,9
         X = 1.
         Y = (I-1)*1.2 + .8
         CALL CALCMP(X,Y,0,1)
         X = 11.
         Y = (I-1)*1.2 + .8
         CALL CALCMP(X,Y,1,1)
         IF (I .EQ. 9) GO TO 20000
         DO 20000 J = 1,18
            N = J
            IF (J .GT. 9) N = J - 9
            XL = 1.
            IF (J .GT. 9) XL = 10.9
            X = XL
            Y = (N/10. + I - 1)*1.2 + .8
            CALL CALCMP(X,Y,0,1)
            X = .1 + XL
            Y = (N/10. + I - 1)*1.2 + .8
            CALL CALCMP(X,Y,1,1)
20000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** INDICATE GEOLAT, INVLAT, MLT
C
d     goto 30000
c
      DO 30000 I = 1, 6
         DO 30000 J = 5, 85, 5
            G = (45-J)*PI/180.-(IPASS-1)*270.*PI/180.
            GJ = 1.*J + 5.
            SG = SIN(G)
            CG = COS(G)
            XL = -ST*SG + CT*CG*SINCL
            YL =  CT*SG + ST*CG*SINCL
            ZL =  CG*CINCL
            RL = SQRT(XL**2 + YL**2)
C
            GLAT = 180.*(ACOS(SQRT(XL**2+YL**2)))/PI
            GLATR = GLAT*PI/180.
            ILAT = 180.*(ACOS((SQRT(RE/(RE+ALT)))*COS(GLATR)))/PI
```

```
C
            MLT = ASIN(YL/RL)*12./PI + 12.
            IF ((XL .LT. 0.) .AND. (YL .GT. 0.))
     1      MLT = ASIN(-XL/RL)*12./PI + 18.
            IF ((XL .LT. 0.) .AND. (YL .LT. 0.))
     1      MLT = ASIN(-YL/RL)*12./PI
            IF ((XL .GT. 0.) .AND. (YL .LT. 0.))
     1      MLT = ASIN(XL/RL)*12./PI + 6.
C
            W = 6. - (G+(IPASS-1)*270.*PI/180.)/.13962641
            IF(I .EQ. 6) GO TO 30010
            H = (I-1)*2.4 + .8
            X = W
            Y = -.05 + H
            CALL CALCMP(X,Y,0,1)
            X = W
            Y = .05 + H
            CALL CALCMP(X,Y,1,1)
            GO TO 30000
30010       IF (ABS(AMOD(GJ,10.)) .GT. 0.) GO TO 30000
            H = .8
            XN = W - .29
            YN = H - .24
            CALL NUMBER(XN,YN,.14,GLAT,0.,2)
            XN = W - .29
            YN = H - .45
            CALL NUMBER(XN,YN,.14,ILAT,0.,2)
            XN = W - .29
            YN = H - .66
            CALL NUMBER(XN,YN,.14,MLT,0.,2)
30000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** DRAW LATITUDE CIRCLES
C
      CNORM=SIN(40.*PI/180.)
c      CALL CALCMP (12.9,8.7,0,3)
      CALL CALCMP (12.7,8.7,0,3)
      CALL CALCMP (0.,0.,0,1)
      CALL GRAIN (0.0)
      IF (IPASS.GT.1) GO TO 04200
      DO 40000 I = 10,40,10
         PRAD=SIN(I*PI/180.)/CNORM
         CALL ARC (PRAD,0.,360.)
40000 CONTINUE
C
C **** OR EQUATORIAL VIEW OF THE EARTH
C
      IF (IPASS.EQ.1) GO TO 05000
04200 CALL CALCMP (X,Y,1,-5)
      DO 42000 I=1,9
         CRAD=COS((I-5)*PI/18.)
         X=SIN((I-5)*PI/18.)/CNORM
         Y=CRAD/CNORM
         CALL CALCMP (X,Y,0,1)
         Y=-CRAD/CNORM
         CALL CALCMP (X,Y,1,1)
42000 CONTINUE
C
      DO 43000 I=10,190,10
         CRAD=COS((I-10)*PI/180.)
         DO 43000 J=1,81
            X=SIN((J-41)*PI/180.)/CNORM
            Y=CRAD*COS((J-41)*PI/180.)/CNORM
            IF (J.EQ.1) CALL CALCMP (X,Y,0,1)
```

```
          CALL CALCMP (X,Y,1,1)
43000 CONTINUE
      CALL CALCMP (X,Y,0,-5)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** INDICATE TIME
C
05000 CALL CALCMP (XORG,YORG,0,3)
      IP=1
      IF (IPASS.GT.1) IP=2
      DO 50000 I = 1, 4
         S = SIN((I-1.)*PI/2.)
         C = COS((I-1.)*PI/2.)
         IF (IPASS.GE.2) GO TO 50001
c          X = .25*S + 12.75
         X = .25*S + 12.55
         Y = -.25*C + 8.6
         CALL CALCMP(X,Y,0,1)
c          X = S + 12.75
         X = S + 12.55
         Y = -C + 8.6
         CALL CALCMP(X,Y,1,1)
50001    RNUM = (I-1)*6.
         GO TO (50010,50020,50030,50040) I
c50010     XL = 12.72
50010    XL = 12.52
         YL = 7.4-(IP-1)*.6
         CALL NUMBER(XL,YL,.14,RNUM,0.,-1)
         GO TO 50000
c50020     XL = 13.82
50020    XL = 13.62
         YL = 8.53
         IF (IPASS.GE.2) GO TO 50021
         CALL NUMBER(XL,YL,.14,RNUM,0.,-1)
         GO TO 50000
50021    CALL SYMBOL (XL,YL,.14,1HN,0.,1)
         GO TO 50000
c50030     XL = 12.63+(IP-1)*.001
50030    XL = 12.43+(IP-1)*.001
         YL = 9.67+(IP-1)*.6
         CALL NUMBER(XL,YL,.14,RNUM,0.,-1)
         GO TO 50000
c50040     XL = 11.45+(IP-1)*.18
50040    XL = 11.25+(IP-1)*.18
         YL = 8.53
         IF (IPASS.GE.2) GO TO 50041
         CALL NUMBER(XL,YL,.14,RNUM,0.,-1)
         GO TO 50000
50041    CALL SYMBOL (XL,YL,.14,1HS,0.,1)
50000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** PLOT ORBIT
C
06000 DO 60000 I = 1, NMEAS
         MP = (1.-(I-1)*2./(NMEAS-1))*PI*40./180.
     1        -(IPASS-1)*270.*PI/180.
         SMP = SIN(MP)
         CMP = COS(MP)
         XL = -ST*SMP + CT*CMP*SINCL
         YL =  CT*SMP + ST*CMP*SINCL
         ZL = CMP*CINCL
c          X=-YL/CNORM+12.75
         X=-YL/CNORM+12.55
```

```
         Y= XL/CNORM+8.6
c         IF (IPASS.GE.2) X=ZL/CNORM+12.75
          IF (IPASS.GE.2) X=ZL/CNORM+12.55
          IF(I .EQ. 1) CALL CALCMP(X,Y,0,1)
          CALL CALCMP(X,Y,1,1)
60000 CONTINUE
C
      IF (MODE .GT. 1) GO TO 07000
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** SHOW X,Y VECTORS
C
c      X = 11.58
      X = 11.38
      Y = 9.07
      CALL CALCMP(X,Y,0,1)
c      X = 11.98
      X = 11.98-0.2
      Y = 9.07
      CALL CALCMP(X,Y,1,1)
c      X = 11.98
      X = 11.98-0.2
      Y = 9.47
      CALL CALCMP(X,Y,1,1)
c      CALL SYMBOL(11.82,9.5,.14,1HX,0.,1)
      CALL SYMBOL(11.82,9.5-0.2,.14,1HX,0.,1)
c      CALL SYMBOL(11.42,9.07,.14,1HY,0.,1)
      CALL SYMBOL(11.42,9.07-0.2,.14,1HY,0.,1)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** INDICATE MAXIMA
C
07000 IPLT=4
      IF (IFLD.EQ.-2) IPLT=3
      IF (IFLD.GT.2) IPLT=3
      DO 70000 I = 1,IPLT
         IF (I.NE.4) GO TO 70005
         IF (FMAX(4).NE.0.) GO TO 70005
         IDEC=-1
         ITY=1
         FACT(I)=1.
         GO TO 70065
70005    IF (I.EQ.4) FMAX(I)=FMAX(I)*1000000.
         IDEC=-1
         IF (FMAX(I).LT.10.) IDEC=1
         IF (FMAX(I).LT.1.) IDEC=2
         IF (FMAX(I).LT..1) IDEC=3
         FACT(I)=1.
70010    IF (FMAX(I).GE.10.) GO TO 70020
         IF (FMAX(I).LT.1.) GO TO 70030
         IF (FMAX(I).GE.1.) GO TO 70040
70020    FACT(I)=FACT(I)/10.
         FMAX(I)=FMAX(I)/10.
         GO TO 70010
70030    FACT(I)=FACT(I)*10.
         FMAX(I)=FMAX(I)*10.
         GO TO 70010
70040    IF (INT(FMAX(I)).LT.INT(FMAX(I)+.5)) GO TO 70050
         FMAX(I)=(AINT(FMAX(I))+.5)/FACT(I)
         GO TO 70060
70050    FMAX(I)=AINT(FMAX(I)+.5)/FACT(I)
C
70060    IF (FMAX(I).LT.1000.) ITY=1
         IF (FMAX(I).LT.100.) ITY=2
```

```
      IF (FMAX(I).LT.10.) ITY=1
      IF (FMAX(I).LT.1.) ITY=1
      IF (FMAX(I).LT..1) ITY=-1
      IF (FMAX(I).LT..01) ITY=-2
70065 XL = 1.2
      YL = (5-I)*2.4 + .4
      N = I + (MODE-1)*4
      CALL SYMBOL (XL,YL,.28,ATEX(N),0.,1)
      IF (I .EQ. 4) CALL SYMBOL (999.,999.,.1,1HV,0.,1)
      XL = ITY*.14+.4
      YL = (5-I)*2.4 + .6
      CALL NUMBER (XL,YL,.14,FMAX(I),0.,IDEC)
      FM = 0.
      XL=.8
      YL = (5-I)*2.4 - .47
      CALL NUMBER (XL,YL,.14,FM,0.,-1)
      FM=-FMAX(I)
      XL=ITY*.14+.26
      IF (FMAX(I).EQ.0.) XL=XL+.14
      YL = (5-I)*2.4 - 1.55
      CALL NUMBER (XL,YL,.14,FM,0.,IDEC)
      XL=.2
      YL=(5-I)*2.4-1.1
      IF (I.EQ.4) GO TO 70070
      CALL SYMBOL (XL,YL,.14,10HNANOTESLAS,90.,10)
      GO TO 70080
70070 YL=YL-.15
      CALL SYMBOL (XL,YL,.14,11HMICROAMPS/M,90.,11)
      XL=XL-.1
      YL=YL+1.54
      CALL SYMBOL (XL,YL,.07,1H2,90.,1)
70080 IF ((I.EQ.4).AND.(FMAX(I).EQ.0.)) GO TO 70000
      IF (I.EQ.4) FMAX(I)=FMAX(I)/1000000.
70000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
C **** PLOT FIELDS
C
    DO 80000 I = 1,IPLT
      IF ((I.EQ.4).AND.(ALT.LT.ALTI)) GO TO 80000

c  don't plot if FMAX(I) = 0.0, otherwise bombs out with FLTDIVZER
      if (fmax(i).eq.0.0) goto 80000

      DO 80000 J = 1,NMEAS
        MP = (J-1)*2./(NMEAS-1)
        X = 5.*MP + 1.
        Y = 2.4*(4-I) + 2. + 1.2*FL(I,J)/FMAX(I)
        IF (J .EQ. 1) CALL CALCMP(X,Y,0,1)
        IF (J .EQ. 1) GO TO 80000
        CALL CALCMP(X,Y,1,1)
80000 CONTINUE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C
    CALL CALCMP(0.,0.,0,3)
    CALL SYMBOL(0.,10.64,.35,21HB-FIELD OF BIRKELAND ,0.,21)
    CALL SYMBOL(999.,999.,.35,13HCURRENT MODEL,0.,13)
    CALL SYMBOL(0.,.66,.14,6HGEOLAT,0.,6)
    CALL SYMBOL(0.,0.45,.14,6HINVLAT,0.,6)
    CALL SYMBOL(0.,0.24,.14,3HMLT,0.,3)
    ORY=7.2
    IF (IPASS.GE.2) ORY=6.5
    CALL SYMBOL(12.55,ORY,.14,5HORBIT,0.,5)
    CALT = ALT/1000.
```

```
      CALL SYMBOL(11.7,6.0,.14,17HALTITUDE =      KM,0.,17)
      CALL NUMBER(13.2,6.0,.14,CALT,0.,-1)
      RCODE = NCODE
      CALL NUMBER(12.,0.,.14,RCODE,0.,0)
      CALL NUMBER(999.,999.,.14,CL1,0.,0)
      CALL NUMBER(999.,999.,.14,CL2,0.,0)
      RUMT=NUMT
      CALL NUMBER(999.,999.,.14,RUMT,0.,0)
      RUML=NUML
      CALL NUMBER(999.,999.,.14,RUML,0.,0)
      RFLD=IFLD
      CALL NUMBER(999.,999.,.14,RFLD,0.,0)
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@)
C

      CALL PAUS
      CALL CALCMP(X,Y,1000,2)
C
      RETURN
      END

      program abrkplt it
        call abrkplt(I,'   ',0)
        stop
      end
```